GLOBAL VIRTUAL VAULT: PREVENTING UNAUTHORIZED PHYSICAL DISCLOSURE BY THE INSIDER

Mike Fisk, Scott Miller, Alex Kent
Los Alamos National Laboratory

ABSTRACT

Information providers on networks such as the Global Information Grid need to share sensitive information while still protecting that information from misuse. We show how common information-sharing mechanisms encourage and allow high-bandwidth, hard-to-detect information exfiltration by malicious insiders, and by adversaries in the field. By leveraging netcentricity, modern stateless clients, and advances in distance visualization techniques, we can provide analysts and warfighters with highly-usable access to information that remains secured in high-availability, high-security data centers. We quantitatively analyze the intentional and inadvertent data exfiltration paths of several off-the-shelf secure computing solutions and demonstrate how to re-engineer these systems to greatly reduce residual risk by limiting access to human-interaction protocols. This approach eliminates large classes of insider attacks that are largely unaddressed in most systems and concentrates traditional insider access to manageable, well-defended physical security perimeters.

I. INTRODUCTION

The implicit trust placed in users with immediate access to computing hardware such as a desktop or laptop is one of the biggest threats to computer and information security. This threat is more commonly referred to as the insider threat. Once such physical access has been granted, the prevention and detection of a user injecting malware or exfiltrating data is difficult if not impossible. While most Internet-connected networks have easier data exfiltration paths than the insider, tightly monitored or air-gaped networks may leave the insider as the simplest data exfiltration path.

In this paper we describe the Global Virtual Vault, a LANL-developed system for providing computing resources to users while minimizing the residual risk involved in granting them access. We minimize this risk by concentrating data storage and processing into a system of physically controlled computing vaults that provide limited, restricted access to the desktop computing environment through a purpose-specific network.

While the user is still granted visual, keyboard, and mouse access to the system, we control their ability to add peripherals or storage or connect to the network and we do so without placing any trust in the desktop hardware or software. By removing dependence on the trust implicit in the end user network and workstations, the major complicating factor in prevention and detection is removed.

In this project we are building unparalleled, systemic controls on the ability of malicious insiders and theives to physically remove information. We are concerned with high-security environments where insiders must be allowed to use large quantities of information, but must be prevented from removing that information. Carnegie Mellon's CERT found that 75% of the proprietary and confidential information thefts they studied between 1996 and 2002 were committed by current employees [11]. While people can always memorize information and leave with it in their heads, there are plenty of situations where there are quantities of sensitive information that are so large that they can only be stolen electronically. Dodge showed that of 2007 incidents at education institutions, 37% were caused by loss or theft of physical storage devices [7]. 2006 data from the Privacy Rights Clearinghouse showed 37% of breaches involved a lost or stolen laptop and another 16% from some other theft [19]. In 2006, the US Department of Veterans Affairs reported that a thief had stolen a laptop containing personal information for about 26.5 million individuals [21]. A class action lawsuit was filed for $1,000 per individual affected [16] and the VA payed millions of dollars in postage and initially offered to pay $160.5 million in credit monitoring services [9].

Here are some other examples of these problems in a variety of industries:

Thanks to Tadeusz Raven, David Sayre, Lynn Saxton, Bill Weiss, Ahmad Douglas, and other participants in the Cyber Futures Lab for their assistance prototyping and testing the Global Virtual Vault and to Roger Hagengruber for supporting our work.

Credit Card Transactions: A Verizon study over 4 years of intrusions shows that 84% of breaches compromised cardholder information [1]. In 2007, a malicious database administrator physically removed and then sold 2.3 million consumer records from Certegy Check Services, including information on 99,000 credit cardholders [4].

Financial Services: In 2006, Fidelity Investments announced the compromise of information belonging to 196,000 retirement-account customers [15].

HIPAA: In 2007, a computer containing information on 38,000 patients at Emory University was stolen from a commercial data processing service [3].

Trade Secrets: In 2007, a former DuPont employee pleaded guilty to downloading 22,000 sensitive documents containing $400 million worth of company trade secrets [11]. Two employees of LG were persuaded to provide thousands of files regarding proprietary production technology for plasma display to an employee of a Chinese competitor. LG predicts a $1.4 billion loss in sales over the next three years as the Chinese competitor brings the technology to market [13].

Defense & Intelligence: A software engineer in the U.S. downloaded 200 classified engineering documents worth $600 million over years from a secure corporate network and attempted to hand-carry those documents to China [20]. Retired Marine and FBI analyst Leando Aragoncillo was convicted of downloading classified information on the Phillipines from FBI computers to disks that he took home and then e-mailed to the Philippines [23]. Randall Craig used a thumb drive to steal and sell Social Security numbers of 17,000 Marines to a person he believed to represent a foreign government [10].

In Section II, we show that these sorts of unauthorized disclosures of information are nearly impossible to prevent when users are given hands-on access to the computers that process this information. However, we also show how the entire end-to-end lifecycle of information can take place under tight controls that dramatically reduce the risk of disclosure while still empowering users to create, manipulate, and view information.

This paper does not directly address the protection of information from remote outsider-oriented threats. For computer systems that are accessible by an outsider — via network access or similar trust relationships — traditional cyber defenses are paramount. But for sensitive information, those defenses are often quite strong and serve at least as a deterrent. In these cases, the insider threat becomes a substantial portion of the residual risk. It is that risk that we are reducing.

II. Insider-Threat Exfiltration from a Standard Desktop Machine

The best opportunity for a malicious insider to steal information is to remove media from their environment. This threat is readily addressed by removing media from the environment; Windows, Unix, and Macintosh systems can all be run without disks using network booting. Network booting prevents the threat of just walking away with the computer or the drive from the computer.

While the system can function without mass storage, this does not prevent a malicious user from adding mass storage (a hard disk, a thumb drive, etc.) or from using an I/O port (e.g. Ethernet, Firewire, USB, serial, parallel) to move data off of the system. Software configuration or agents are the typical mechanism for limiting storage and I/O peripherals. These agents can disable drivers, log/report the connection of any unapproved peripheral or media, but can be readily bypassed through a variety of paths as follows; local physical access can always defeat local software protections.

First, boot the desktop into an environment with no agent. A firmware password can be used to prevent altering which media can serve as a boot device, but this password does nothing to authenticate whether the boot device is approved or malicious and, if the user has physical access to the machine, can be locally reset. If a machine is configured to boot from internal hard disk, the local disk can simply be replaced.

Second, a rogue computer with its own boot process can be used instead of an authorized computer. This rogue device can either be used to access the network be employed as a rogue server on the network capable of intercepting and hijacking all data including network boot processes, network file shares, and other user sessions. In an environment that restricts access to certain MAC addresses, the user would have to change MAC address of the rogue client, a trivial task. With more sophisticated network access control systems that require cryptographic authentication of the computer using a protocol like IEEE 802.1x, the user might have to obtain additional access credentials. If these credentials are stored on the authorized system's hard disk, it can be trivially read from the disk by the rogue computer. In principle, storing the credential in a Trusted Platform Module (TPM) [14] on the authorized client would protect the credential. However, proper (secure) use of a TPM is quite problematic since it requires binding the credential to the boot loader, the OS, kernel modules and drivers and other security-critical components. Every permutation of every upgrade of these components must be accounted for. Because of this complexity, truly secure TPM use is still far from widespread.

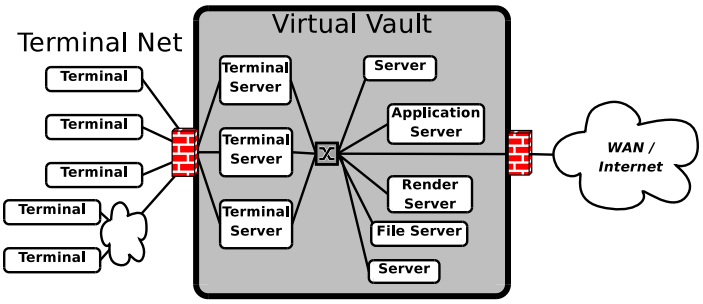

Fig. 1: Global Virtual Vault Architecture

Third, aside from the modifying the bootstrapping process, most contemporary operating systems and desktop environments are also vulnerable to privilege escalation attacks by an authorized user without modifying the bootstrapping process.

Fourth and finally, use locally enabled ports to exfiltrate data. As described in Section V, the DVI port on a computer cannot be disabled in most cases and provides a very high bandwidth channel that can be recorded. If the attacker can add an expansion card to a PCI bus — a port present on the inside of virtually any modern computer, workstation, or commodity-based thin client — the attacker can examine physical memory out of the system [5]. Worse still, on modern laptops with ExpressCard slots, the PCI bus is an external connector. External FireWire ports also provide direct access to memory on many systems [8], [2].

Because of these known vulnerabilities and the general complexity of most desktop computers and their software (which all but guarantees additional vulnerabilities only one of which need be successful and useful), we do not view it as tractable to control a general purpose computer from having recordable output. Instead, we propose an architecture and a threat model that does not require us to trust the desktop computer.

III. The Virtual Vault Architecture

Our metaphor for secure computing is the glovebox — not the glovebox in your car, but the kind of glovebox used to allow experimenters to manipulate hazardous materials. Figure 2 shows such a glovebox and illustrates the concept: a transparent air-tight container with gloves that allow the user to reach into the container and manipulate what is inside without having direct contact with what is inside the box.

The computing analog of this architecture separates the user's physical computing environment (i.e. office) from the secured internal environment. Figure 1 shows this architecture. All sensitive processing should be limited to the secured environment. For example, the user's desktop system should never have access to sensitive data or run applications on that data. In fact, we follow the principle of least privilege and grant the user only the functional access that they require: video output and keyboard and mouse input. We adopt the name *terminal* access to describe this human input and output access since it is reminiscent of the serial terminals of older mainframe and supercomputer systems. We will show in later sections how modern technology successfully delivers a contemporary, graphics-rich environment to these terminals.

Fig. 2: A glovebox used to contain dangerous material in laboratory environments

The objective of this separation between processing and users is to control data exfiltration. The specific policy of our architecture is the following:

> ***Only trusted processes in the internal environment should be allowed to communicate with the user-accessible external network.***

Most rich-content environments and operating systems are the antithesis of this policy, allowing for immediate, dynamic instantion and linking of a wide variety of libraries and applications. With everything linked transparently together, a direct attack is not necessary; only one of these libaries or applications needs to contain a useful vulnerability that would allow for data exfiltration. In particular, the graphics display system itself posses capabilities that are excellent for data exfiltration: large bandwidth, large memories, ready access from user programs, and connection to a high-bandwidth output port. Although these vulnerabilities are common to all modern operating systems – Windows, Unix and Unix-like environments, OSX, etc. – we will focus momentarily on Unix and Unix-like systems as a specific, well-described example of the problem-at-hand.

In X Windows, the display system of most Unix and Unix-like systems, the display is managed by the X Server, which means it traditionally runs in the user's physical environment. Any application can be a client of that server and generate output to display and receive input via the server. If our user is using an X-Windows Terminal, then the internal environment has to allow any of the user's processes to connect to the X Server on the terminal. This violates our separation between environments.

Since the user's environment is not trustworthy, consider the case where the X Server port on the terminal simply accepts connections and saves all data to removable storage. Any application in the internal environment can exfiltrate any sort or volume of information simply by opening a TCP connection to the X Server.

For a more subtle attack, the inside application can load

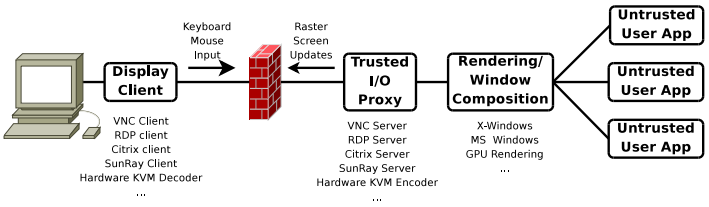

Fig. 3: Display Architecture

undisplayed pixmaps in to a real X Server. These pixmaps can be raw data instead of an actual image. Another X client in the user's environment can grab these pixmaps and save them to storage. In this way, the data exfiltration can use an unmodified X Server, but does not even need to be a visible process.

In contrast, consider a display system such as Microsoft's Remote Desktop Protocol (RDP) or the VNC protocol[18] which introduces another component, the display client, as shown in Figure 3. Typically, only the client runs in the user's environment. The server runs on the internal environment and the applications display to it (typically the VNC server reads already-rendered bitmaps from a traditional display system such as X-Windows or from the graphics hardware). Thus, the only network connectivity required between the internal and user environments is between the client and the server, the RDP or VNC server acting as a trusted proxy between user applications and the user display. Thus, client applications are limited to exfiltrating data as actual, displayed video display information which can be strictly rate-limited and monitored. We explore those concepts in Section V.

IV. Virtual Vault Network Architecture

A. Trusted Network Processes

We previously stated that only trusted processes should be allowed to communicate with the user network. In this section, we describe how one can implement this policy for a given trusted process.

Local, host-based firewalls such as Linux iptables or many Windows host-based firewalls can limit the network access of specific users and processes. Thus, for a given trusted process, we can insert a host-based firewall rule that allows that process to communicate. For example, to instantiate a trusted VNC server, a process with permission to insert firewall rules (e.g. a root process) should launch the VNC server and then insert a rule that allows that process to talk. As the parent process, it will know when the VNC server terminates and can remove the rule before the PID is recycled. In environments such as BSD or Solaris that lack firewalling by PID, the parent process can wait for the VNC server to bind to a network port and then insert a rule that applies to that specific port. In Windows environments, local firewall policy can only allow Remote Desktop or similar to communicate. As a single trusted process relays all network traffic, the network need only support a small subset of protocols. Thus, network-based firewalls and ACL's can be deployed as additional protection layer.

Unfortunately, many RDP, VNC, and similar implementations lack strong authentication and encryption. We have developed a VNC session management system that uses SSH running over a non-standard port. Only this port is allowed to communicate between the user network and the internal network. This SSH server is configured with *ForceCommand* so that all connections execute the session manager process. The session manager instantiates a new VNC server or identifies an existing one that is suitable. It then opens a connection to that server and relays traffic to the client. It is worth noting that a normally-configured SSH server can be run simultaneously for communication within the internal network. This demonstrates the ability to armor data-in-transit and force strong authentication of the user before any server access is granted.

B. Terminal Isolation

There is always a risk that one terminal may be compromised by another system on the terminal network. For example ARP pollution attacks can be used to launch man-in-the-middle attacks on terminal on the same network (using tools like *dsniff* or *ettercap*). Since these terminals are usually diskless devices that boot from the network, there is also the risk of creating a rogue boot server.

Fortunately, many switch vendors implement ARP and DHCP snooping in which all packets (including the payload in ARP responses) from edge ports are dropped if they contain an IP or MAC address other than what was assigned to that port by a trusted DHCP server [22].

To mitigate the risk of one edge port attacking other terminals, we configure the terminal local area network to isolate the terminals from each other. We do this using Ethernet switch filters or with "isolation groups." With the combination of only allowing trusted processes to connect from the server back to untrusted clients, strong network authentication to validate users, encryption to protect data in transit, and enforced isolation between the clients allowing only communication directly to the server, the number of exfiltration channels is bounded.

V. Video Exfiltration

We have stated that we are accepting the risk of providing video display output to the user. At some point, this is a fundamental assumption of general-purpose computing environments. However, video output has a significant

bandwidth and it can be recorded. So have we gained anything by reducing our exposure to this level? This section answers that question in the affirmative.

The Transition Minimized Differential Signaling specification used by DVI and HDMI is 165 million 24-bit pixels per second for a total of nearly 4 gigabits/second [6]. However, much of this bandwidth is used for blanking intervals and other overhead. The practical bandwidth limit is 2.4 megapixels at 24 bits per pixed and 60Hz, which still works out to more than 3 gigabits/second of bandwidth. Clearly, video signals provide a huge bandwidth for moving information. In a traditional computing environment, only the speed of the graphics hardware and system I/O busses limit the ability to use this entire bandwidth. Further, off-the-shelf hardware exists from companies like NCast, Foresight Imaging, Epiphan, and EMS Imaging to capture DVI signals and record them to disk.

However, our terminal video protocols typically run at well under 100 megabits/second thanks to sophisticated encoding techniques. In fact, many are usable over 56 kilobit/second or 1 megabit/second wide-area links. These efficient protocols make extensive use of compression and transmitting only changed portions of a screen. This encoding means that we can drive a full 3GB/s DVI display effectively with on the order of a 1/1000th the bandwidth. More importantly, we can enforce a maximum bandwidth and use that enforcement as a tight upper-bound on the video output channel bandwidth. Thus, we can limit the potential video display channel exfiltration. In a highest-security environment, we can limit the video protocol to 56 kilobits/second and bound the potential loss to be more than 5 orders of magnitude less than the raw capacity of a DVI channel. Even in a graphics-intensive environment, we can limit the channel bandwidth by a factor of 50.

This tight rate-limiting and quantification of the video bandwidth is a significant achievement. Further, we expose this video protocol to additional profiling, auditing, and quota enforcement. For example, we could define that most users have not just a burst limit, but also a per-day bandwidth that is just sufficient to support their normal bandwidth usage integrated over a whole day.

Finally, we can chose to perform anomaly detection on video protocol usage. Especially in environments where static quotas may impact work, anomaly detection can easily identify (e.g. through time-series change detection algorithms) increases in activity over normal levels. This increased activity can be investigated to identify abuse.

A. Video Performance

Our architecture depends on satisfying users' graphics and video expectations while encoding all video output in relatively low-bandwidth network streams. Contemporary desktop environments make extensive use of streaming video and hardware-accelerated 3D rendering.

Benchmark	Local	TurboVNC	Remote X
sw	16.74	9.49	11.98
tcvis	3.83	3.83	3.81
3dsmax	14.61	10.88	0.51
catia	18.04	3.38	0.73
maya	20.61	11.33	0.60
proe	10.56	3.63	0.42

Fig. 4: Video Rendering Performance (frames/sec)

In our open systems implementation, we use VirtualGL, a system that uses graphics hardware on the computer where the application is actually running and then reads-back the rendered result as a raster image which it then displays to the client [17]. VirtualGL was developed for distance visualization environments and wide area networks. VirtualGL can work with any X display client and is also optimized to work with TurboVNC.

In fact, our tests show that VirtualGL performance outperforms 3D rendering the data on a diskless client in many cases. Unless data is already stored locally on a thick client, it often takes more bandwidth to transfer the raw data to the client for rendering than to transfer already-rendered data from a render server. Figure 4 shows SPECviewperf 10.0 benchmarks on three different platforms: a traditional disk-full system; a VirtualGL and TurboVNC-based remote display traversing WiFi, VPN, ADSL, and 110ms round-trip times; and a 100 Mbps-limited LTSP terminal (a remote X display where the application runs on the disk-full system and the rendering occurs on tht terminal, which is connected by a 100Mbps LAN). Values are frame rates and higher is better. While the disk-full system does have better numbers, the VirtualGL client was quite usuable, even over a slow WAN. Meanwhile, the remote X display was all but useless even over a LAN due to X's extreme sensitivity to bandwidth and even small latency.

Citrix has also optimized DirectX and streaming video performance from Windows terminal servers by pre-rendering on the server and then sending compressed, streaming video over the network to the client [12].

VI. Printing Exfiltration

So far we have assumed that video output on a monitor is the only output to users. While many work environments can function without hardcopy, some environments still require the ability to print. Our approach to printing is to provide auditability and to allow bandwidth limitation similar to what we have done for video.

First, we use printing systems that require the user to authenticate themselves at the printer itself before it will

print their jobs. This protection prevents another user from picking up the job from a printer before the originating user gets to the printer.

Second, we can enforce printing quotas based on print job size rather than page count. The print job size is the more accurate representation of the information content of the job. For example, a plain text, large-font print job may consume many pages with only a few kilobytes of information. However, an 8.5x11 inch document on a 600dpi monochrome laser printer can contain 30 Megabytes of information per page.

Many print jobs contain images and graphs at a higher resolution than can be printed. In the future, we plan to provide mechanisms to automatically downsample these jobs to the maximum quality of the printer or do a predefined, visually acceptable resolution. By peforming this down-sampling, we can reduce the size of user quotas without impacting their actual ability to print.

VII. Inside the Vault

We use the term Global Virtual Vault because users can be anywhere in the world and the data is still protected. However, we assume that the internal network is indeed in a physically secure environment. In our implementation, we have engineered a secure co-location facility.

The facility is accredited for unattended storage of classified matter. Access control systems require two authorized individuals and three-factor authentication (badge, PIN, and hand geometry) for access. The access control system also enforces that at least two authorized users are inside at all times. Authorized staff are subject to special screening.

Within the facility, servers are protected similarly to bank safe deposit boxes. Servers are kept in custom-made, locked co-location racks and the vault security officer does not have access to the keys for these racks. The only people who may have keys for a rack compartment are individuals who are privileged users for all of the machines in that compartment.

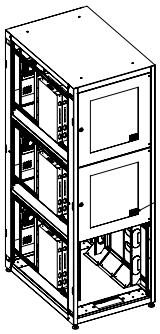

Fig. 5: Co-location racks.

These co-location racks are divided into three separate compartments. Each compartment is fully enclosed in sheet metal (with perforated doors for ventilation) and a high-end lock with interchangeable core. There is a separate cableway for each compartment that extends to ceiling and floor for either under-floor or overhead cabling.

VIII. Related Work

A. Rights Management

Digital Rights Management (DRM) is terminology used to describe technology that controls how a user can use data. Movie and music content providers rely heavily on DRM to control unauthorized duplication of material. DRM technologies are rarely very secure since they typically lack any trustworthy environment to operate in. For example, DVD content can be encrypted but the lack of any key distribution infrastructure means that every DVD player had to have a static decryption key. Some DVD player hardware in computers displayed the movie directly to the screen without allowing software to ever have a copy of the unencrypted content. But eventually software DVD players were released containing the decryption algorithm and key. These players were promptly reverse engineered to yield the algorithm and the key (which is trivially brute-forced once the algorithm is known).

The Trusted Platform Module (TPM) [14] is a secure co-processor available on many PCs today. The TPM provides secure storage and use of cryptographic credentials. These credentials can be stored by a running computing environment with the caveat that they only be made accessible to the same environment. Environments are authenticated through a chain of cryptographic hashes of their memory. Thus, a secure bootstrapping of a system can occur. However, most contemporary systems are instantiated with so many permutations of versions of software components, that it has yet to be made practical to perform such a bootstrapping. In a simpler environment with restricted functionality and more homogeneity (e.g. network-booted instances of a special-purpose thin client system), this secure bootstrapping is much more approachable.

B. Thin Clients

There are many off-the-shelf thin client systems in use today. We break these clients into two classes: local-booting and network-booting. Local-booting thin clients contain an embedded operating system (typically Windows XP Embedded, Windows CE, or Linux) which is stored in flash memory. This flash memory is frequently provided in the form of an IDE storage device that functions like a small hard disk. Many operating environments make this storage device writable by users. From a security perspective, these devices are equivalent to a traditional PC. While network-booting thin clients do not come with their own mass storage, they are equivalent in both functionality and security to a diskless PC described above. In summary, thin clients exist because of their ease of deployment and reduced hardware costs, but do not offer any fundamental security improvements over other PCs and workstations.

C. Hardware Remote KVM

Hardware KVM extenders allow the Keyboard, Video, and Mouse (KVM) devices to be physically remote from the

computer. Historically, these extenders used closed circuits and proprietary links. For example, the ClearCube C/Port uses a dedicated fiber between each computer and user. These proprietary links have distance limitations of a few hundred meters or less, which means that users must be physically near the computers. For our goal of providing access anywhere in the world over a variety of network technologies, this limitation is unworkable.

Newer hardware KVM extenders use IP over Ethernet and therefore remove distance limitations. However, they still dedicate a whole computer to each user. These systems are usable in our environment and fit our security requirements. For example, the ClearCube I/Port does hardware filtering of USB traffic and can restrict it to just mouse and keyboard input. Video output is compressed raster images and can withstand the same bandwidth limitations described in Section V.

D. Multi-level Ultra-thin Clients

Ultra-thin client systems have received multi-level accreditation for preserving the separation between classification levels. Some systems perform that isolation using a multi-level terminal server. Others push multi-level operation all the way out to the client. In both cases, the only processing being done is terminal server clients. However, these systems keep the terminals and terminal network accredited at the highest level in the system and do specifically prevent the exfiltration of data to the terminals.

IX. Conclusions

In this paper we have presented our novel network, system, and security architecture for managing and greatly reducing the threat posed by physical access to computers. This architecture addresses both insider threat concerns as well as physical loss and theft concerns. We present it as an important complement to the body of work in network and system security architectures that mitigate the threat posed by network-borne attacks. True security cannot be achieved without adressing both network and physical threats.

Our contributions are the exfiltration-based analysis of protocols and architectures that allow authorized data movement, including on-screen video, to be proxied by trusted processes without to give user desktops access to sensitive networks.

Our experience deploying and operating this environment with multiple instantiations of technology — both open-source based and proprietary; both Windows and Unix — has strengthened our confidence in this architecture. We have shown that video performance is better than other, less secure architectures and that this architecture also lets us enjoy the efficiences in cost of ownership provided by server-based computing.

A. Future Work

We are currently working to add support for secure, strongly authenticated telephony to this environment. We are also persuing ways ot alter both video and printed imags in imperceptible ways to further limit the effective bandwidth of exfiltration mechansisms.

References

[1] Wade Baker, C. David Hylender, and J. Andrew Valentine, "2008 data breach investigations report," Tech. Rep., Verizon Business RISK Team, 2008.

[2] Adam Boileau, "Hit by a bus: Physical access attacks with firewire," *Ruxcon 2006*, 2006, (talk).

[3] Martin H. Bosworth, "Emory healthcare laptop stolen," *Consumer-Affairs.com*, Jan. 2007.

[4] Jim Carr, "Fidelity: Employee stole, sold 2.3 million consumer records," *SC Magazine*, July 2007.

[5] Brian D. Carrier and Joe Grand, "A hardware-based memory acquisition procedure for digital investigations," *Digital Investigation*, vol. 1, no. 1, pp. 50–60, 2004.

[6] "Digital video interface," Tech. Rep., Digital Display Working Group, 1999.

[7] Adam Dodge, "Educational security incidents (ESI) year in review – 2007," Tech. Rep., Feb. 2008.

[8] Maximillian Dornseif, "0wned by an iPod," *PacSec 2004*, 2004, (talk).

[9] John Files, "V.A. laptop is recovered, its data intact," *New York Times*, June 2006.

[10] Grant Goss, "Military computer contractor pleads guilty to id theft," *Computer World*, May 2008.

[11] Larry Greenemeier, "Massive insider breach at dupont," *Information Week Online*, Feb. 2007.

[12] AutoDesk, Inc., "Autodesk joins Citrix alliance program and announces AutoCAD map 3d software is 'Citrix ready'," (press release), Apr. 2008.

[13] H. Kim, "Former LG Electronics man charged with spying for china," *Yonhap News Agency*, Mar. 2008.

[14] Chris Mitchell, *Trusted Computing*, IET, 2005.

[15] Associated Press, "Data breach at fidelity," *New York Times*, Mar. 2006.

[16] Associated Press, "Data on 2.2M active troops stolen from VA," *USA Today*, June 2006.

[17] The VirtualGL Project, "A study of the performance of virtualgl 2.1 and turbovnc 0.4," Tech. Rep., May 2008.

[18] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper, "Virtual network computing," *Internet Computing, IEEE*, vol. 2, no. 1, pp. 33–38, 1998.

[19] Beth Rosenberg, "Chronology of data breaches 2006: Analysis," Tech. Rep., Privacy Rights Clearinghouse, Feb. 2007.

[20] Ashok Selvam, "Worker accused of stealing secrets," *Chicago Daily Herald*, Apr. 2008.

[21] David Stout, "Personal data of 26.5 million veterans stolen," *New York Times*, May 2006.

[22] Cisco Systems, *Catalyst 6500 Series Switch Cisco IOS Software Configuration Guide*, chapter Configuring DHCP Snooping, 2008.

[23] District of New Jersey U.S. Attorney's Office, "FBI intelligence analyst arrested, accused of passing classified information; former philippines national police official also charged," (press release), Sept. 2005.